LAMDA at TREC CDS track 2015

Clinical Decision Support Track

Moon Soo Cha, Woo-Jin Han, Garam Lee, Minsung Kim, Kyung-Ah Sohn*
Department of Information and Computer Engineering
Ajou University
Suwon, Republic of Korea
(ckanstnzja; data; piratekl; kimmsql; kasohn)@ajou.ac.kr

***Abstract*— In TREC 2015 Clinical Decision Support Track, our goal is to retrieve the relevant medical articles for the questions about medical statement. We propose three main strategies of indexing, query expansion, and the ranking method. In the indexing stage, each medical article is indexed into 3 different fields: title, abstract, and body. Before querying, related words are appended to the query at the query expansion stage. Our system returns the score of each field corresponding to the query for all documents. The score of each field is calculated using Divergence-from-randomness (DFR) probabilistic model. With the 3 scores from each field, the total score is calculated as the weighted sum of each score. Finally, we pick up top 1000 documents and send the list of the articles for evaluation. To make it easier for building the IR system, Elasticsearch and MetaMap are adopted for general IR operations and query expansion, respectively. Elasticsearch supports the similarity module that defines how matching documents are scored. In our IR system, Divergence-from-randomness model is adopted for probabilistic term vector space model because it is figured out that DFR outperforms all the other vector space models supported by Elasticsearch. MetaMap is the online tool that maps biomedical text to the Metathesaurus, and its semantic type. Query expansion is executed by extracting the semantic type from the description of the question, and appending words in the same semantic types to the query.**

Keywords—TREC, Retrieve, DFR, MetaMap, Clinical Decision Support, Query expansion, Elasticsearch

I. INTRODUCTION

The clinical decision support system has been continuously required for developing the linking system between medical cases and relevant information [1, 2]. In TREC 2015 Clinical Decision Support Track, our goal is to retrieve the relevant medical knowledge from the query, which is from medical cases. The medical knowledge consists of 700,000 biomedical documents supported by the PubMed Central [3] which is online digital database freely available. A medical case is made of two different fields: description and summary, and each medical case belongs to one of 30 different topics. The topics are divided into 3 types: diagnosis, test, and treatment. In Task B, additional information is provided with a diagnosis field for the test and treatment type.

Fig. 1 shows the overview of our system that consists of both our clinical decision support system and Elasticsearch framework [4]. Our system is made up of data converting, indexing, query expansion and ranking method. Data conversion module parses documents and stores them in database. Next, indexing module handles tokenization and constructs the index database using Elasticsearch. When a doctor requests relevant information for the medical cases, query expansion adds the additional information with medical case in query. As a result, we get the relevant information ordered by score using similarity model.

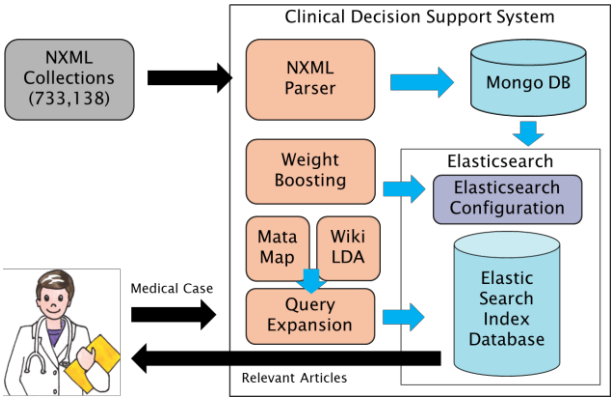

Figure 1. System Overview

II. METHOD

A. Data Converting

NXML file as medical knowledge contains full texts of each document that is XML encoded using the NLM Journal Archiving and Interchange Tag Library. Then, NXML Parser using both XML Path Language (Xpath) and Document Type Definition (DTD) is built for extracting the information of pmcid, author, title, abstract, and body of the medical knowledge. The information is stored using MongoDB.

B. Indexing

We have tried to make an experiment with two tokenization methods, Unicode text segmentation and Edge N-gram. Unicode text segmentation is standard tokenizer supported from Elasticsearch. But it recognizes similar words differently. For example, if there exists the "obesityA" and "obesityB" word, it can recognize the same word or other word. So, we used Edge N-gram per segmentation word that is similar to N-gram but only generates N-gram from the beginning of the word. If we search "obesity" word, it can recognize "obesityA" and "obesityB".

To index the medical knowledge, Elasticsearch framework is adopted for building the information retrieval system with

tokenization. Each document is indexed into three different fields: title, abstract and body.

C. Query Expansion

A medical case, which is query in our case, is made up of two types, description and summary. We evaluated simply the precision and recall between description and summary as query. As a result, we found better performance using summary than description [5].

Query expansion is the process of reformulating a query to improve retrieval performance in information retrieval. We developed two methods for query expansion. First, we expand the query from each description. That is, we extracted words as semantic types including Body Part, Organ, Organ Component, Disease Syndrome, Pharmacologic Substance, Sign Symptom, Diagnostic Procedure and Finding using MetaMap [6] in description. Note that MetaMap discovers metathesaurus concepts referred to in text and divides 133 medical semantic types. Then, if there are given diagnosis fields in Task B, we used Latent Dirichlet Allocation (LDA) [7] based query expansion in Wikipedia. We follow five steps. First of all, we found the diagnosis Wikipedia page and downloaded the Wikipedia page as pdf. Then, we extracted text using PDFBOX Library [8] and word as semantic types in text using MetaMap. Finally, we selected top twenty words using LDA as query expansion word.

D. Ranking method

1) Similarity Model

Elasticsearch provides the various similarity models including TFIDF cosine similarity, BM25, Divergence From Randomness (DFR), Information Based (IB), Language Model with Dirichlet similarity (LMD) and Language Model with Jelinek Mercer similarity (LMJ). To select the similarity model, we constructed each index per similarity model and tested the precision and recall through 2014 TREC CDS track. Then, we selected Divergence From Randomness (DFR) model [9].

DFR model is one type of probabilistic model. It consists of both randomness model P_m and the first normalization P_{risk}. Randomness model is global representation of the term using geometric approximation of Bose-Einstein model and the first normalization is local representation using normalization H2. It follows the term weighting (1).

$$\mathrm{w(t, D)} = -P_{risk}(d_t|D) \log P_m(d_t|C) \quad (1)$$

2) Weight Boosting

For scoring the retrieved documents, we adopted weight boosting method to optimize the score weight of each field. We trained the weight of each field with 2014 TREC Clinical Decision Support Track Result Data. Weight boosting method [10] used the least square error between the relevance score and the expected score for training weights of each fields. Note that the relevance score is the answer for 2014 TREC CDS track and the expected score is a value resulted from our system. We found the optimized weight of field through greedy search.

3) Borda Fuse Scoring Method

In Task B, we used Borda Fuse Ranking Score model [11, 12] that is based on election strategies such as voting model. If there are two Model A and B, Ranking Score follows formula (2).

$$\text{Ranking Score} = \frac{1}{Rank_A + Rank_B} \quad (2)$$

We used two models between query expansion using description and only additional diagnosis word as query. Then, we calculated re-ranking score from ranking result in two models.

III. Results

A. Submitted Run

Our submitted runs are described Table 1. There were total 6 runs for Task A and B, three runs per each Task. Lamdarun01 was summited using DFR Model and standard tokenization with summary query and description query expansion. Lamdarun02 was only different tokenization as Edge-N-gram with lamdarun01. Weight boosting was added to lamdarun03 that was assigned different weights to title, body and abstract.

Task	Run ID	Method
A	lamdarun01	- DFR Model - Summary as Query - Standard tokenizer - Description Query Expansion
	lamdarun02	- DFR Model - Summary as Query - Edge N-gram - Description Query Expansion
	lamdarun03	- DFR Model - Summary as Query - Standard tokenizer - Description Query Expansion - Weight Boosting
B	lamdarun04	- DFR Model - Summary as Query - Standard tokenizer - Description Query Expansion with Additional Information - Weight Boosting
	lamdarun05	- DFR Model - Summary as Query - Standard tokenizer - Description Query Expansion - Wikipedia Query Expansion - Weight Boosting
	lamdarun06	- Borda Fuse (Run 01 + Only Additional diagnosis field as Query)

Table 1 Description of Submitted Runs

In Task B, diagnosis terms were used to retrieve the articles. The suggested diagnosis terms were added to a query expansion in lamdarum04. In the lamdarun05, we extracted important terms from Wikipedia with diagnosis terms and added to query expansion. In the last run in Task B, Borda Fuse

method was used for ranking between added diagnosis information lamdarun01 and retrieved articles with only diagnosis term for query.

B. Evaluation

The performance of our submitted runs is evaluated based on the official result from 2015 TREC CDS Track. There are four measures for evaluation: infAP, infNDCG, R-Prec, and P@10. In the first and the second run, standard tokenizers supported from Elasticsearch and Edge N-Gram are compared. It is observed that Edge N-gram does not contribute to performance improvement. In the third run, our system with weight boosting and stardard tokenizer is evaluated. The performance is slightly raised in terms of infNDCG, R-Prec and P@10.

Run ID	infAP	infNDCG	R-Prec	P@10
Median	0.0414	0.2038	0.1615	0.3433
lamdarun01	0.0364	0.1798	0.1513	0.3133
lamdarun02	0.0364	0.1798	0.1507	0.3133
lamdarun03	0.0363	0.1811	0.1519	0.3167

Table 2 Evaluation on Task A

In Task B, our system with query expansion and Borda Fuse ranking model shows similar performance improvement as Median. In the last run, Borda Fuse ranking model is evaluated, and it has no contribution to the performance.

Comparing the result of our 6 runs with Median, all of the results in Task B are better than Task A. Therefore, a diagnosis field of task B is effective in improving the performance. In Task B, the performance is enhanced without use of diagnosis field. Thus, depending on whether the diagnosis field is used for querying, the result of Task B increases significantly.

Run ID	infAP	infNDCG	R-Prec	P@10
Median	0.0633	0.2794	0.2123	0.45
lamdarun04	0.0508	0.2337	0.1862	0.3867
lamdarun05	0.0657	0.2758	0.2228	0.4267
lamdarun06	0.0656	0.2708	0.2129	0.44

Table 3 Evaluation on Task B

IV. Conclusion

In this work, we proposed three strategies to build IR system for clinical decision support: indexing, query expansion, and the ranking method. We have evaluated 6 runs across 2 Tasks. In Task A, we have run our system with Edge N-gram, compared to the one with weight boosting and standard tokenizer. Our system with Edge N-gram shows no performance improvement in terms of infNDCG, R precision, and P@10. In Task B, we have evaluated our system in query expansion stage. Our system with query expansion using Wikipedia performs better than the one only with description. In conclusion, we found that standard tokenizer in the indexing stage, weight boosting in document scoring stage, and query expansion using Wikipedia performs the best over our trials.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Science, ICT, and Future Planning (MSIP) (2014R1A1A3051169 & 2010-0028631)

References

[1] K. Roberts, M. Simpson, D. Demner-Fushman, E. Voorhees, and W. Hersh, "State-of-the-art in biomedical literature retrieval for clinical cases: a survey of the TREC 2014 CDS track," *Information Retrieval Journal*, pp. 1-36.

[2] M. S. Simpson and D. Demner-Fushman, "Biomedical text mining: a survey of recent progress," in *Mining text data*, ed: Springer, 2012, pp. 465-517.

[3] M. S. Simpson, E. Voorhees, and W. Hersh, "Overview of the TREC 2014 Clinical Decision Support Track," in *Proc. 23rd Text Retrieval Conference (TREC 2014). National Institute of Standards and Technology (NIST)*, 2014.

[4] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide*: " O'Reilly Media, Inc.", 2015.

[5] A. Tombros and M. Sanderson, "Advantages of query biased summaries in information retrieval," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 2-1

[6] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: historical perspective and recent advances," *Journal of the American Medical Informatics Association*, vol. 17, pp. 229-236, 2010.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003.

[8] J. P. PDFBox, "processing Library," *Link: http://www.pdfbox.org*, 2014.

[9] G. Amati and C. J. Van Rijsbergen, "Probabilistic models of information retrieval based on measuring the divergence from randomness," *ACM Transactions on Information Systems (TOIS)*, vol. 20, pp. 357-389, 2002.

[10] J. I. Garcia-Gathrighta, F. Menga, and W. Hsua, "UCLA at TREC 2014 Clinical Decision Support Track: Exploring Language Models, Query Expansion, and Boosting."

[11] J. A. Aslam and M. Montague, "Models for metasearch," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 276-284.

[12] S. Choi and J. Choi, "SNUMedinfo at TREC CDS track 2014: Medical case-based retrieval task."